



INDUSTRIAL AND MANAGEMENT
ENGINEERING, POSTECH



머신러닝 기반 스마트폰 파지 자세 탐지 및 구분 모델 개발

POSTECH
POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

 INDUSTRIAL AND MANAGEMENT
ENGINEERING, POSTECH

 Ergonomic Design
Technology Lab



미래산업
사용성평가센터
Usability Center

권도훈, 최신아, 유희천

포항공과대학교
산업경영공학과
인간공학설계기술 연구실

본 연구는 산업통상자원부의 "미래첨단 사용자편의서비스 기반조성사업"의 지원을 받아 수행된 연구결과임 (R0004840, 2020)

Contents

- 서론
 - 연구 배경
 - 연구 목적
- 문헌조사: 머신러닝 기반 이미지 탐지/구분
- 연구 방법
 - 스마트폰 파지 자세 데이터 획득 실험
 - 데이터 전처리/모델 개발
- 연구 결과
 - 정확도/분류 결과
- 토의

제품 파지 자세와 사용성

- 제품의 부적절한 user interface(UI)는 **불편한 파지 자세를 유도**할 수 있음
 - ✓ 조작하는 대상의 위치에 따라 파지 자세가 달라짐(Wobbrock et al., 2008)
 - ✓ 파지 자세에 따라 상호작용 방법이 달라짐(Yang et al., 2012)
- 불편한 파지 자세는 **사용성 저하, 근피로도 증가, 기기 slip을 유발**함

제품의 다양한 파지 자세



불편한 파지 자세로 인한 부정적 영향



Slip 발생



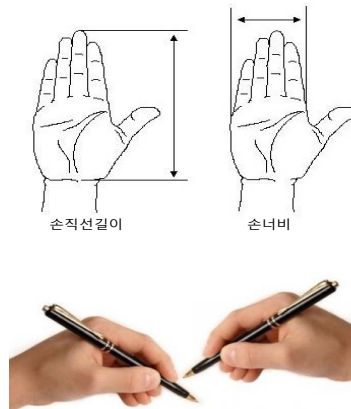
스마트폰 파지 자세 파악 필요성과 한계

- 다양한 사용자의 **선호 스마트폰 파지 자세를 파악**하여 UI를 설계한다면 **사용성을 향상** 시킬 것으로 기대됨
- 현재 스마트폰 파지 자세 파악은 **분석자의 수기/직접 관찰**로 이루어지고 있음
⇒ **많은 시간과 노력이 필요함**

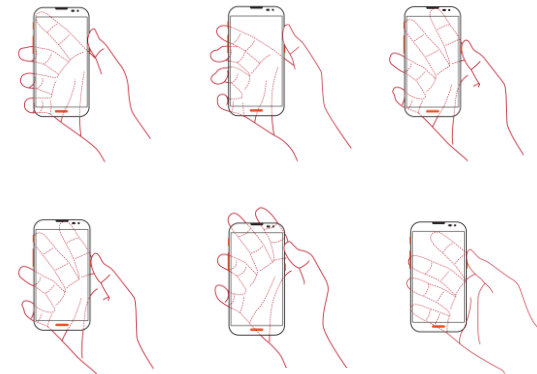
크기 별 스마트폰



인체측정학적 요소



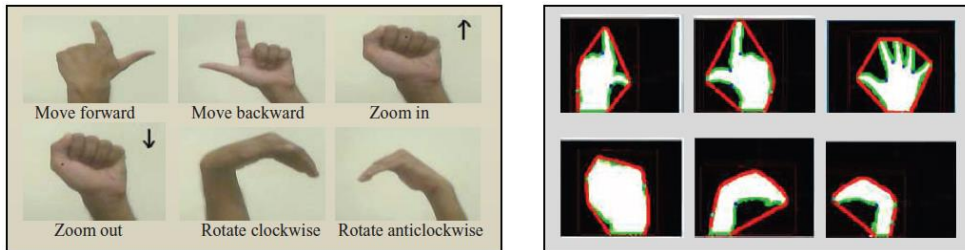
선호 파지 자세 파악



파지 자세 파악의 머신러닝 기법 활용성

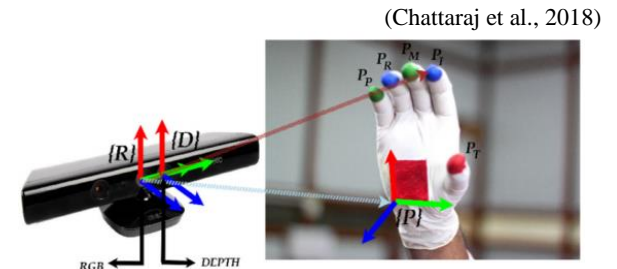
- 사용 자세 파악에는 **장비 및 머신러닝 기법**을 통해 **이미지/영상에서 손을 탐지하는 hand tracking 기술**이 사용될 수 있음
- 제품 사용시에는 자연스러운 **손 움직임을 방해**하지 않는 **이미지/영상 기반 hand detection 기술**이 더 적절하게 활용될 수 있음

Image recognition



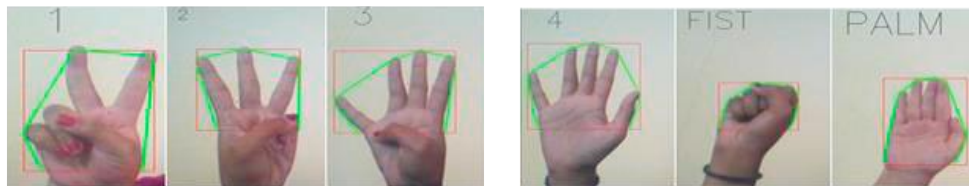
(Rautaray and Agrawal, 2012)

Kinect, glove

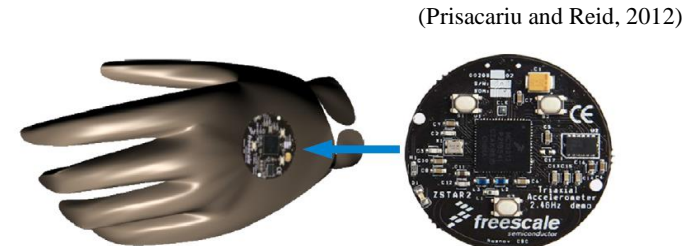


(Chattaraj et al., 2018)

Accelerometer



(Haria et al., 2017)



(Prisacariu and Reid, 2012)

머신러닝 기법을 활용하여 스마트폰 파지 자세 탐지 및 구분

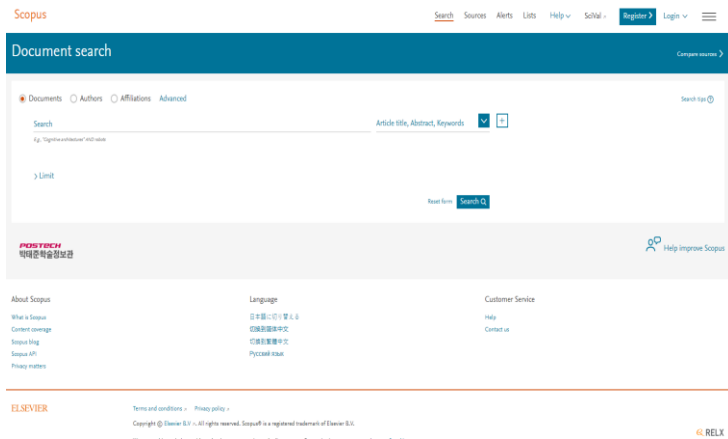
1. Hand tracking 문헌 조사
2. 스마트폰 파지 자세 평가 및 분석 protocol 정립
3. 파지 자세 데이터 전처리 및 hand detection model 개발
4. 스마트폰 파지 자세 classification model 개발

문헌 조사

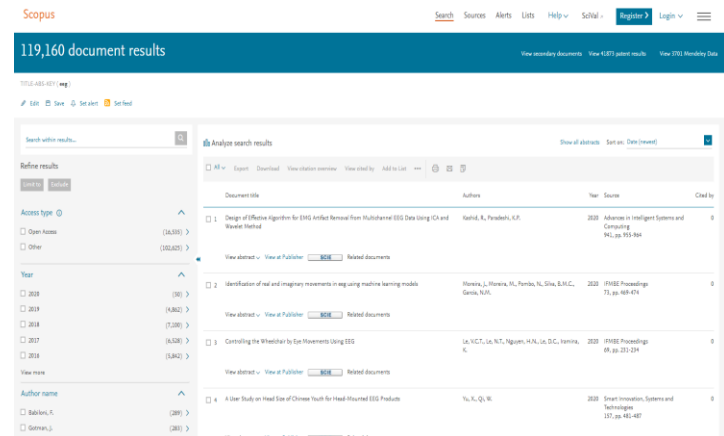
문헌 조사 방법

- ❑ 문헌 조사 site: sciencedirect.com, scopus.com
- ❑ 검색 조건: title, abstract, keyword
- ❑ 검색 keyword
 - ✓ **Hand tracking** 관련: recognition, tracking, detection, classification
 - ✓ **파지 자세** 관련: hand, hand posture, grip posture, grasp

문헌 검색 site: scopus.com



문헌 검색 결과 예시



문헌 선정 절차

S1. **Keywords 조합**을 통한 journal paper 검색

TITLE-ABS-KEY

(("recognition" OR "detection" OR "tracking"
OR "classification") AND ("hand" OR "grip
posture" OR "grasp" OR "hand posture"))

→ **1394 건**

S2. **Title screening**을 통한 1차 선별



→ **401 건**

S3. **Abstract screening**을 통한 2차 선별

→ **176 건**

S4. 입수된 full paper에 대한 **관련도 평가**

관련도 평가 결과

상: 35건

중: 70건

하: 115건

S5. 관련도에 따라 **최종 review 대상 논문** 선별

상: 35건 review

문헌 선정 결과



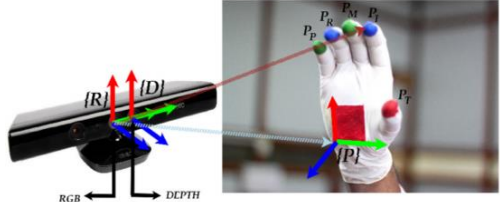
□ 선별된 문헌들의 기술개발목적, 사용하는 device

No.	Year	Title	Authors	Affiliation	Journal	Objective	Device	Model/image base	Algorithm
1	2011	Feature extraction from hand image based on new detection operator	Zhiquan Feng, Bo Yang, Yuehui Chen, Yanwei Zheng, Tao Xu, Yi Li, Ting Xu, Deling Zhu	Ji Wei Road 106, Jinan, Shandong, China School of Information Science and Engineering, University of Jinan, Jinan 250022, PR China Provincial Key Laboratory for Network Based Intelligent Computing, University of Jinan, Jinan 250022, PR China	Pattern Recognition	Gesture recognition	2D camera	Image base	Coarse location for feature – polygon boundaries 이용하여 배경에서 손 검출 후 link model 생성
2	2007	A real-time hand tracker using variable-length Markov models of behaviour	Nikolay Stefanov, Aphrodite Galata, Roger Hubbard	Advanced Interfaces Group, School of Computer Science, University of Manchester, Manchester M13 9PL, UK	Computer Vision and Image Understanding	Gesture recognition	2D camera	Image base	Gaussian mixtures algorithm (Stauffer and Grimson)을 이용하여 손의 hue value를 기준으로 배경과 손(low hue)을 분리 후 손을 polygon으로 단순화 시켜 관절 각도 검출
3	2000	Gesture recognition using the multi-PDM method and hidden Markov model	Chung-Lin Huang, Ming-Shan Wu, Sheng-Hung Jeng	Institute of Electrical Engineering, National Tsing-Hua University, Hsin-Chu, Taiwan, ROC	Image and Vision Computing	Gesture recognition	2D camera	Image base	PDM과 Hidden Markov Model(HMM)을 결합한 검출 알고리즘: 추가 조사 필요
4	2010	A variational approach to monocular hand-pose estimation	Martin de La Gorce, Nikos Paragios	MAS Laboratory, Ecole Centrale de Paris, Grande Voie des Vignes, 92 295 Chateay-Malabry, France	Computer Vision and Image Understanding	Gesture recognition	2D camera	Model base	Hand silhouette 검출 후 각 관절 모델 분석하여 자세 추정
5	2011	A multi-view vision-based hand motion capturing system	Meng-Fen Ho, Chuan-Yu Tseng, Cheng-Chang Lien, Chung-Lin Huang	Institute of Electrical Engineering, National Tsing Hua University, HsinChu, Taiwan, ROC Department of electronic Engineering, Hsiuping Institute of Technology, Taichung, Taiwan, ROC Department of Computer Science and Information Engineering, Chung-Hua University, HsinChu, Taiwan, ROC	Pattern Recognition	Gesture recognition	2D camera, depth camera	추가 조사 필요	배경 분리, 각 관절 각도 계산, multi-view data 활용, separable state based particle filtering (SSBPF) 추가 조사 필요
6	2012	3D hand tracking for human computer interaction	Victor Adrian Prisacariu, Ian Reid	Department of Engineering Science, University of Oxford, United Kingdom	Image and Vision Computing	Gesture recognition	2D camera, accelerometer	Image base	Hand 검출 후 가속도센서 값 활용하여 정확한 rotation 적용, Visual 3D tracker 정보 추가 조사 필요
7	2004	Model-based visual hand posture tracking for guiding a dexterous robotic hand	Jinshi Cui, Zengqi Sun	State Key Laboratory of Intelligent Technology and Systems, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China	Optics Communications	Gesture recognition	2D camera	Model base	Hand model 검출 시 PF (particle Filter) 와 GA (genetic algorithm) 을 융합하여 GA기반 particle filter를 개발
8	2012	Gravity optimised particle filter for hand tracking	Malik Morshidi, Tardi Tjahjadi	School of Engineering, University of Warwick, Gibbet Hill Road, Coventry CV4 7AL, United Kingdom	Pattern Recognition	Gesture recognition	2D camera	Model base	PF를 개선한 GOPF 방법으로 검출
9	2017	Non-parametric hand pose estimation with object context	Javier Romero, Hedvig Kjellström, Carl Henrik Ek, Danica Kragic	Perceiving Systems Department, Max Planck Institute for Intelligent Systems, 72076 Tübingen, Germany CVAP/CAS, KTH, SE-100 44 Stockholm, Sweden	Computer Vision and Image Understanding	Hand pose in product use	2D	Image base	물건을 쥐고 있는 hand tracking 기술과 기존 기술 비교
10	2017	Guided optimisation through classification and regression for hand pose estimation	Philip Krejov, Andrew Gilbert, Richard Bowden	Centre for Vision, Speech and Signal Processing, Faculty of Engineering and Physical Sciences, University of Surrey, United Kingdom	Computer Vision and Image Understanding	Gesture recognition	Depth camera	Image base	Voxel grid filter를 이용해 point cloud data에서 손 형상 분석
11	2008	Hand gesture recognition and tracking based on distributed locally linear embedding	S.S. Ge, Y. Yang, T.H. Lee	Social Robotics Lab, Interactive Digital Media Institute & Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576	Image and Vision Computing	Gesture recognition	2D camera	Model base	Distributed locally linear embedding (DLLE), modified version of the locally linearly embedding (LLE), Probabilistic neural network



문헌 조사: 실험 장비

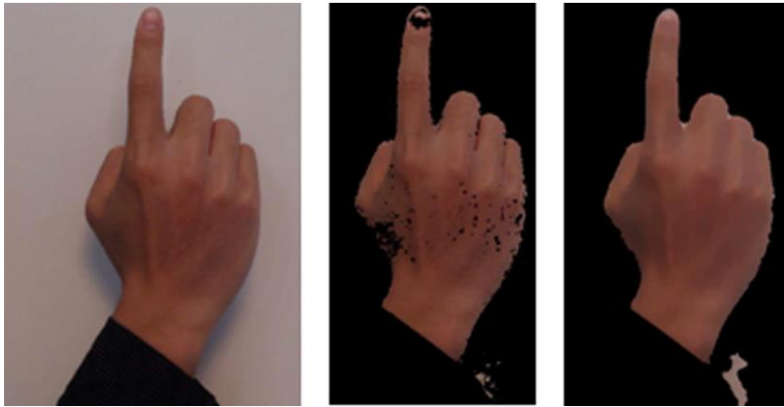
- 측정 방법에 따라 (1) 2D camera, (2) depth camera, (3) glove 방식으로 구분
- Camera 기반 방법은 참여자의 자연스러운 손 움직임에 방해가 되지 않음

	2D camera	Depth camera	Glove
설명	일반적인 video camera 기반 녹화	이미지 각 픽셀 값의 깊이를 계산	Marker가 부착된 glove를 통해 손 모양 측정
사진			
장점	비용과 노력 ↓ 자연스러운 손 움직임 측정 가능	3차원 형태 파악 가능	Marker를 통해 link 생성 ⇒ 정확한 손 모델 구현 가능
단점	심층적 분석 불가능, noise 영향	장비와 프로그램 필요	비용과 노력 ↑, 손 움직임 방해 발생
사용 빈도	XX% (20/35건)	12.5% (5/35건)	37.5% (10/35건)

문헌 조사: 손 추적 Algorithm

- Hand tracking은 공통적으로 (1) capturing, (2) finding, (3) analyzing의 3단계를 거치며, **hand shape**을 분석하는 **image 기반**과 **3D configuration**을 분석하는 **model 기반**으로 구분됨

Capturing and finding



(Premaratne et al., 2017)



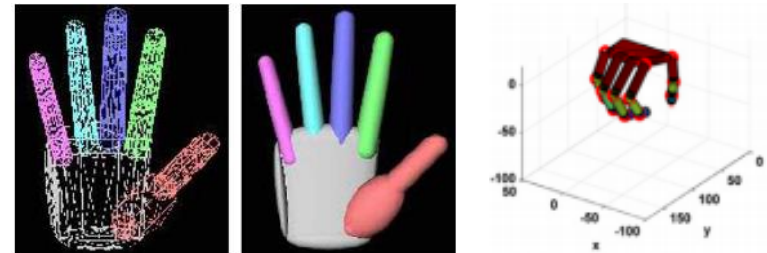
(Stefanov et al., 2007)

Image based analyzing



(Lee et al., 2013)

Model based analyzing



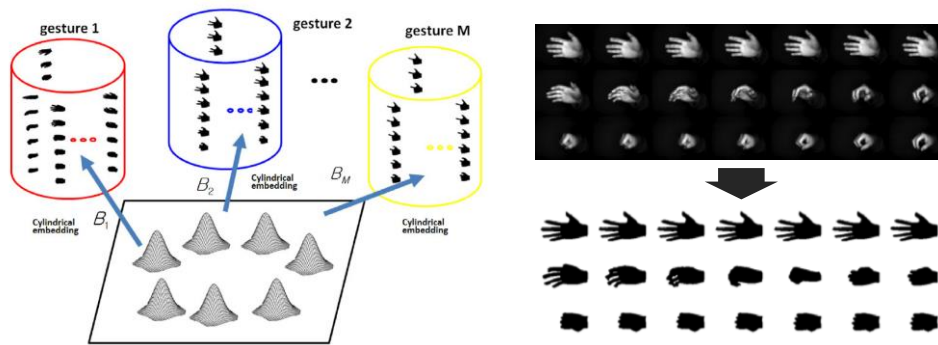
(Cui and Sun, 2004)

(Chattaraj et al., 2018)
Ergonomic Design
Technology Lab

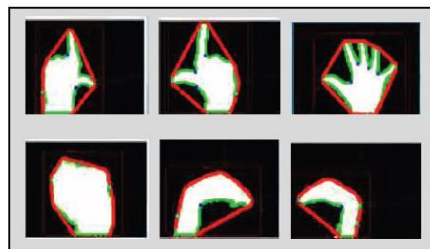
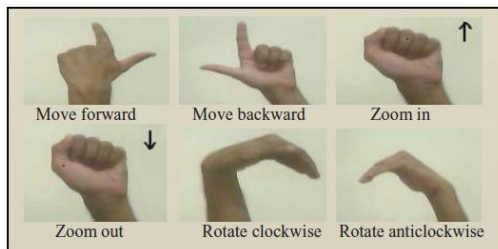
문헌 조사: 손 추적 Algorithm: Image 기반(1/2)

- 각 hand pose에 대한 다수의 Image를 촬영 혹은 생성하여 학습시킨 후 분석 대상의 hand pose를 분석
- 손의 세부적인 자세보다는 손의 움직임을 추적하여 gesture를 인식하는 연구들이 많음

Hand pose analysis

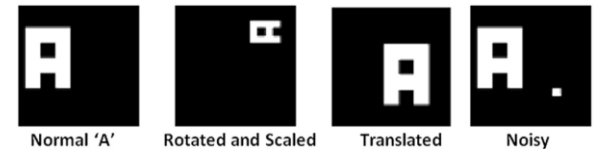


(Lee et al., 2013)

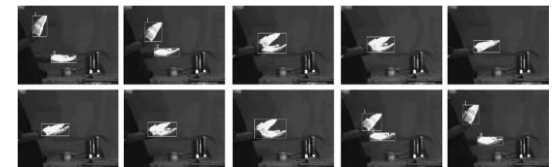


(Rautaray and Agrawal, 2012)

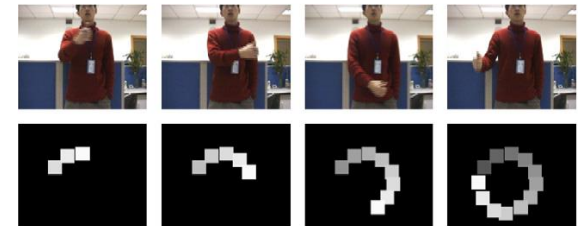
Hand movement tracking



(Premaratne et al., 2013)



(Shamaie and Sutherland, 2005)



(Shan et al., 2007)

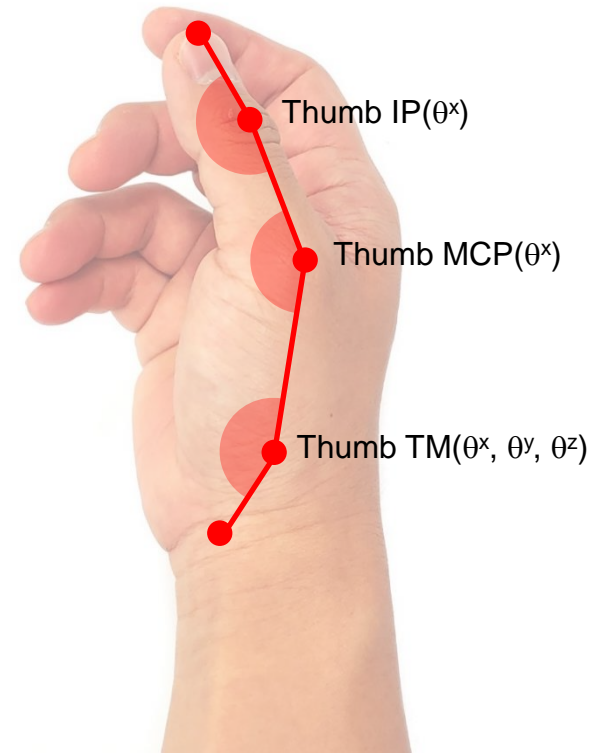
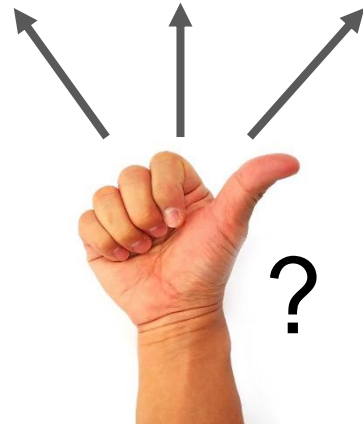
문헌 조사: 손 추적 Algorithm: Image 기반(2/2)

- 학습된 pose 외의 새로운 hand pose에 대한 분석이 불가능함
- 각 손가락의 세부적인 위치 및 각도 분석이 제한됨

학습용 hand pose



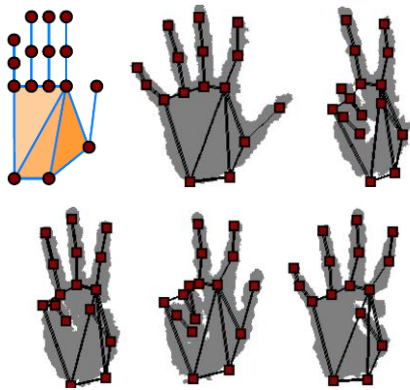
분석 대상 hand pose



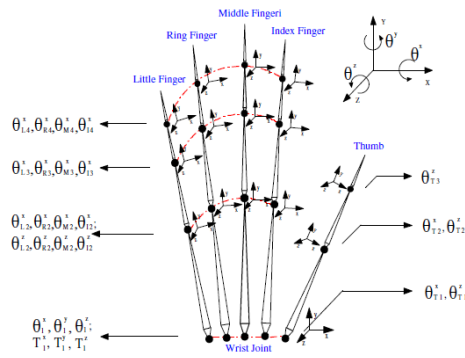
문헌 조사: 손 추적 Algorithm: Model 기반(1/2)

- Hand kinematic model을 **link의 수와 joint의 자유도를 연구 목적에 적절한 수준으로 생성**하여 측정 대상 hand data를 fitting

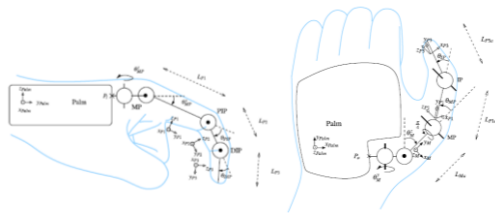
Hand kinematic models



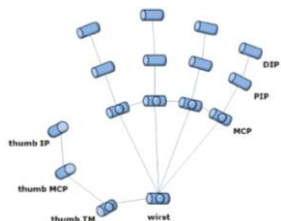
Coleca et al. (2015)



Ge and Lee (2008)

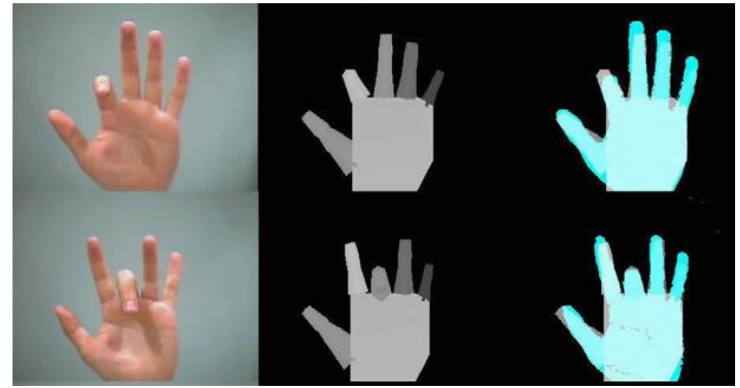


Gorce and Paragios (2010)

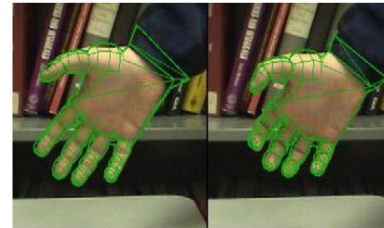


Cerulo et al. (2017)

Fitted hands



Ho et al. (2011)



Gorce and Paragios (2010)



Morshidi and Tjahjaji (2012)

문헌 조사: 손 추적 Algorithm: Model 기반(2/2)

- 제품을 파지한 자세를 분석하기 위해서는 **다방면에서 촬영한 3차원 형상 필요**
- 빠른 조작 상황에 대한 분석은 높은 computing power를 요구함

Hand poses without a product



Hand poses with a product



스마트폰 파지자세 파악 실험

실험 참여자

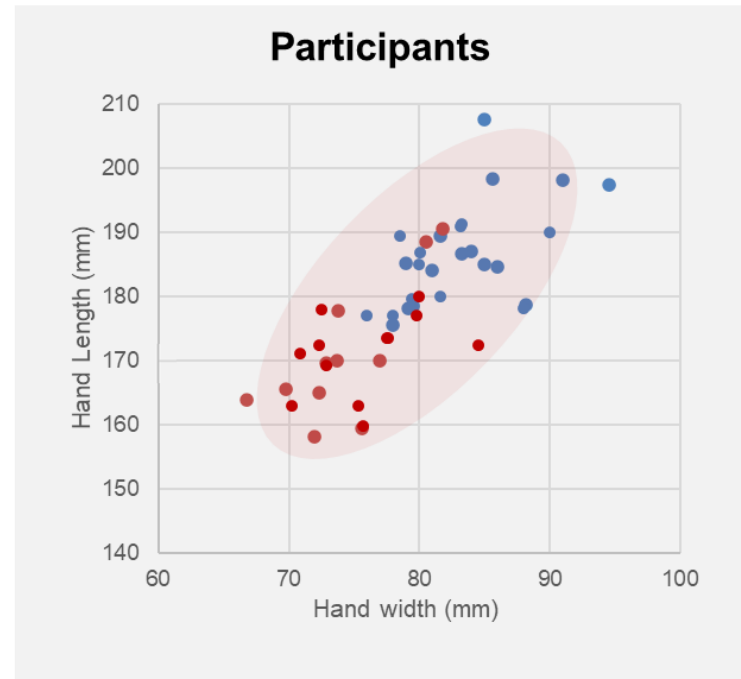
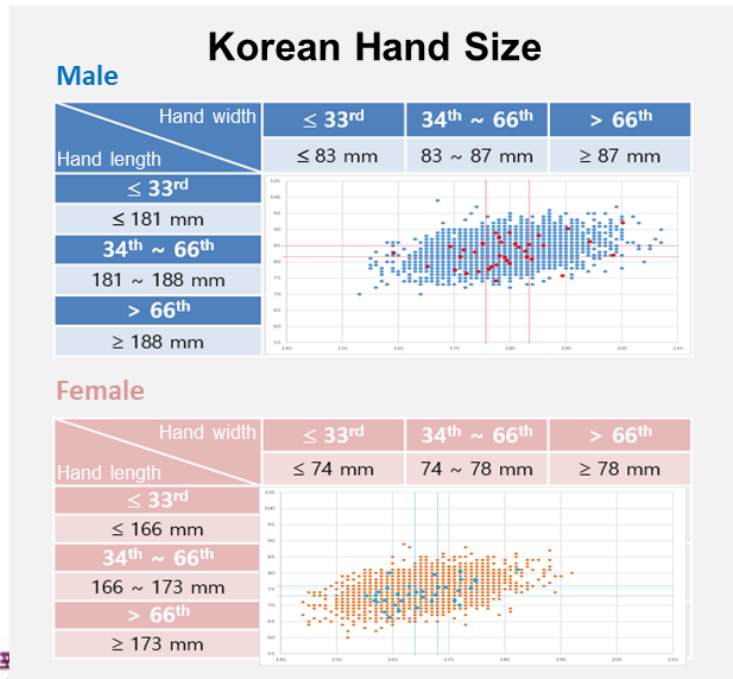
□ $n = 70$ (left-handed: 20; right-handed: 50; female: 35; male: 35)

➤ 연령: 28.8 ± 12.8 yrs (range = 24 ~ 54 yrs)

□ 실험 참여자 모집 기준

➤ 스마트폰 사용 경험 > 3년

➤ 한국인의 손 크기(손 직선 길이, 손 너비) 분포를 고려해 9개 그룹으로 모집



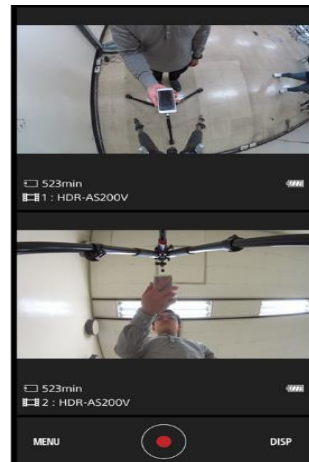
평가 장비

- ❑ **크기와 무게가 다른 스마트폰 mock-up 9종** ← 국내외 스마트폰 설계 제원 참조
 - ✓ 크기: 3.0" ~ 7.0" (0.5" 간격)
 - ✓ 무게: 100 g ~ 190 g (10 g 간격)
- ❑ **사용 작업 측정을 위한 web camera 2대**

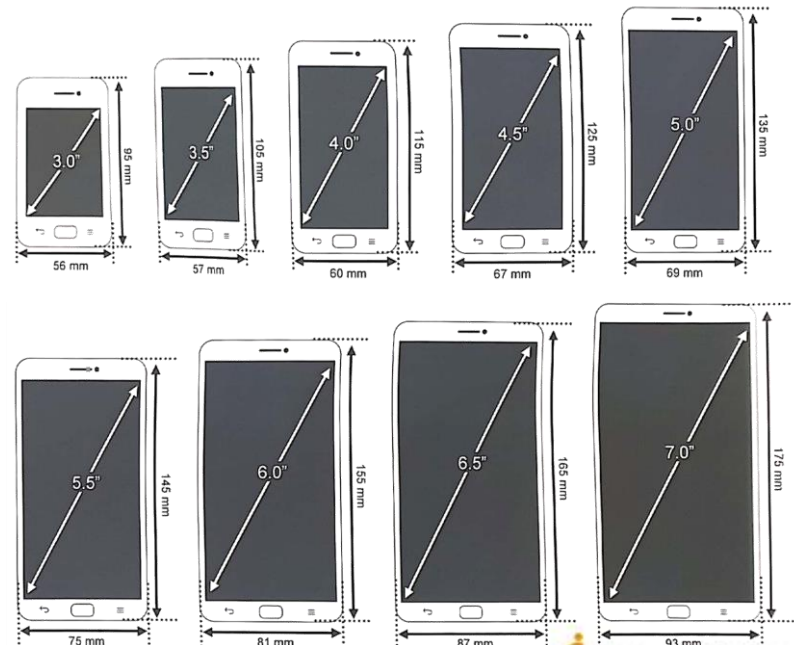
Video camera



촬영 확인용 application

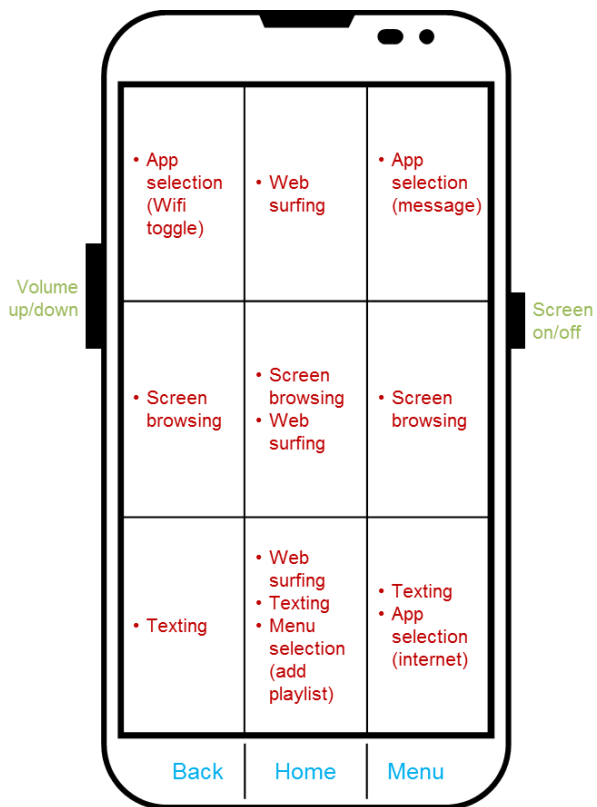


스마트폰 mock-up



평가 작업

- 선행 연구를 통해 스마트폰 사용 중 주요 task와 세부 동작을 도출
- 관련성과 실험실 상황을 고려하여 (1) **Texting**, (2) **web surfing**, (3) **Listening**, (4) **calling**의 4가지 task를 선정함



Tasks	Specific actions
Answering a call	<ol style="list-style-type: none"> 1. Grasp the phone 2. Answer a call by flicking the screen 3. Turn volume up/down by volume key
Listening to music	<ol style="list-style-type: none"> 1. Turn volume up/down by volume key 2. Scroll up/down 3. Show menus 4. Select a menu 5. Turn volume up/down by volume key
Texting	<ol style="list-style-type: none"> 1. Turn screen on by power key 2. Navigate screens 3. Select a message app 4. Send a message 5. Return home
Browsing the web	<ol style="list-style-type: none"> 1. Turn screen on by power key 2. Turn Wi-Fi on/off 3. Select a web browser app 4. Browse the internet 5. Turn screen off by power key

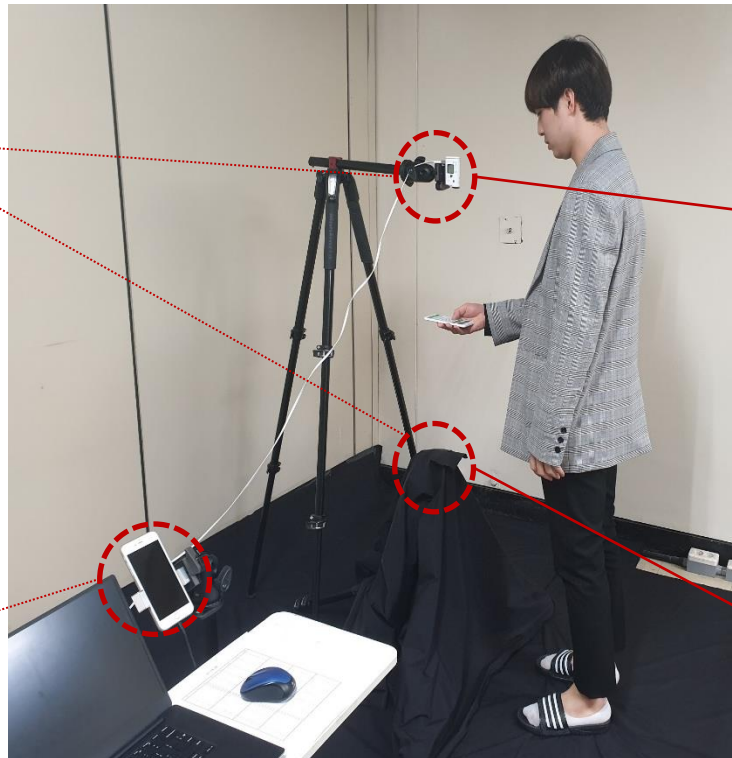
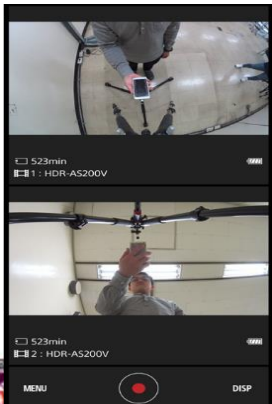
파지 자세 측정 환경

- ❑ **Video camera** (HDR-AS200, Sony Inc., Japan)을 실험참여자의 자연스러운 손 움직임을 방해하지 않도록 **손 위치 영역 상단과 하단에 배치**
- ❑ Hand detection을 위해 **카메라 촬영 영역에 검은 천막을 설치함**

Video camera



App(Imaging edge mobile)을 통한 촬영 확인



Top camera



Bottom camera



실험 절차

소요시간: 총 60분

S1. 실험 설명(10 min)

S2. 실험참여자 손 길이 측정(5 min)

S3. 평가 task 수행 연습(5 min)

S4. 평가 task 수행(15 min)

S5. 휴식 및 debriefing (10 min)

- 실험 목적 및 방법 설명
- 실험 참여 동의서 작성
- 손 길이: 손목 선 ~ 중지 끝 점 사이 거리
- 실험참여자 **우세 손 확인**
- 스마트폰 **mock-up 친숙화**
- **Random 순서**로 task 수행
- **피로 영향 제거** 위해 휴식 제공

오른손 → 왼손
왼손 → 오른손

파지 자세 분석: 작업별 Image Capture

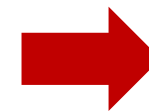
- 파지 자세가 고정된 구간에 대해 **task 수행 단계별로 image capture**
 - ✓ 총 이미지 수: 1931 개(40명 × 9개 mock-up × 10 단계 task)
 - ✓ 연구자 2인 검토 후 파지 자세 고정 구간 확인

Image capture 시점

Display 사용 시



Hard key 사용 시



고정 구간 적합성
Double check

파지 자세 분석: 유형 분류

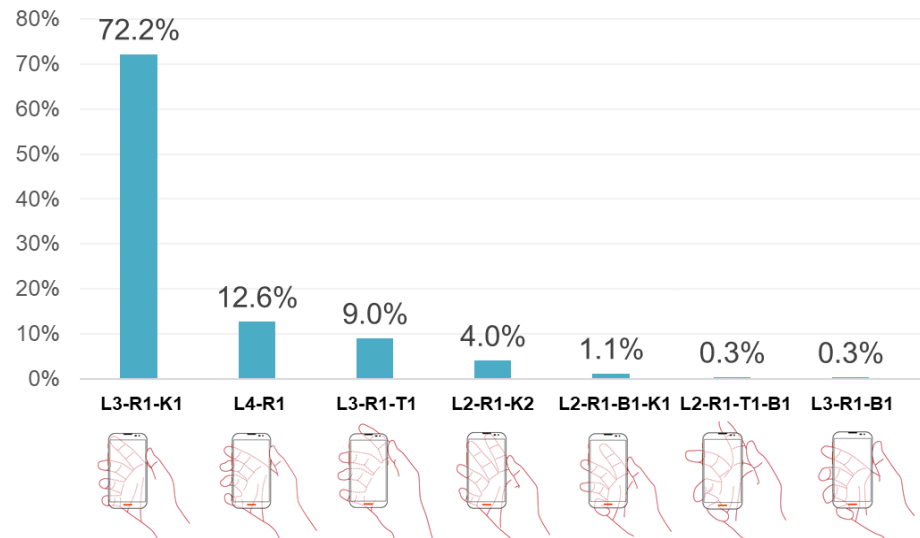
- Mock-up의 각 부위별 위치한 **손가락의 수를 기반으로 파지 자세 유형 분류**
- Smartphone PUI task 조작 시 주로 사용된 **7가지 파지 자세에 대해 분류**

파지 자세 유형 분류 예시: L3-R1-K1



L3-R1-K1

PUI 조작 시 사용되는 grip posture별 사용 빈도



Location	Left (L)	Right (R)	Top (T)	Bottom (B)	Front (F)	Rear (K)
# of fingers	3	1	0	0	0	1

Dataset

- ❑ Input data: grip posture 측정 실험을 통해 얻은 7 types image data
- ❑ **왼손 데이터의 경우 좌우 반전**을 통해 오른손 데이터와 동일한 데이터로 인식

7가지 유형의 파지 자세 데이터 수

No.	Type	Amount
1	L2-R1-K2	175
2	L2-B1-R1-K1	205
3	L2-T1-B1-R1	185
4	L3-R1-K1	754
5	L3-B1-R1	218
6	L3-R1-T1	229
7	L4-R1	165
	Total	1931

같은 유형의 왼손과 오른손의 파지 자세

Left hand



Right hand



Preprocessing

- Capture된 image를 바탕으로 image preprocessing을 수행
 - (1) **Binarization**, (2) **skin color distinguishing** method의 두가지 방법 중 정확도가 더 높은 방법으로 수행
 - ✓ Binarization: 주변 noise를 더 잘 제거하지만, 간혹 손가락을 잘 detection 하지 못하는 경우도 있음
 - ✓ Skin color 기반: 더 정확하지만, 옷 색깔과 같은 noise가 발생함

Raw data와 preprocessing 결과 image

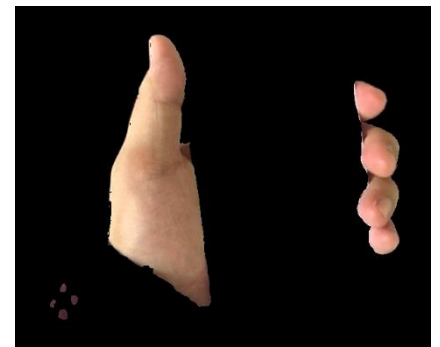
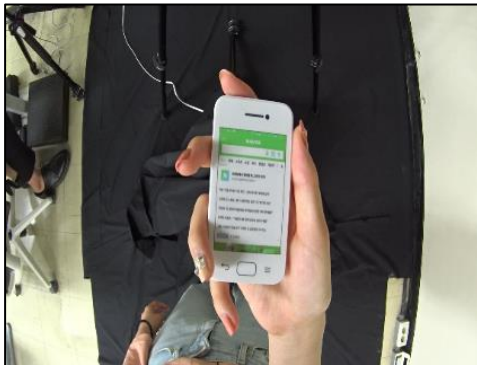
2가지 preprocessing method

Raw data

After preprocessing

Binarization

skin color distinguishing



Preprocessing: 알고리즘 수행 결과 화면

The image displays a Windows file explorer window on the left and a PyCharm IDE on the right. The file explorer shows a directory structure for 'Train_New' with subfolders like 'Grip_Posture' and 'Raw_data_direct'. The PyCharm IDE shows a Python script named 'data_preprocess.py' with the following code:

```
skinroi = skin1[0:skin1.shape[0]-1, skin1.shape[1]/5:skin1.shape[1]/5*4]
imgroi = image[0:skin1.shape[0]-1, skin1.shape[1]/5:skin1.shape[1]/5*4]
contours, hierarchy = cv2.findContours(skinroi, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
xmin = skinroi.shape[1]
ymin = skinroi.shape[0]
xmax = 0
ymax = 0
for i in range(0, len(contours)):
    x, y, w, h = cv2.boundingRect(contours[i])
    xmin = x if x < xmin else xmin
    ymin = y if y < ymin else ymin
    xmax = max(x+w, xmax)
    ymax = max(y+h, ymax)
h = ymax-ymin
w = xmax-xmin
ymin = max(0, ymin-h//10)
xmin = max(0, xmin-w//10)
ymax = min(height, ymax+h//10)
xmax = min(width, xmax+h//10)
crop = skinroi[ymin:ymax, xmin:xmax]
orcrop = imgroi[ymin:ymax, xmin:xmax]
skin = cv2.bitwise_and(orcrop, orcrop, mask=crop)
return crop, orcrop, skin

def run(path, spath1, spath2, spath3):
    if not os.path.exists(spath1):
        os.makedirs(spath1)
    if not os.path.exists(spath2):
        os.makedirs(spath2)
    if not os.path.exists(spath3):
        os.makedirs(spath3)
    for index in os.listdir(path):
        for filename in os.listdir(os.path.join(path, index)):
            if filename.endswith('.png'):
                image = cv2.imread(os.path.join(path, index, filename))
                crop, orcrop, skin = segment_hand(image)
                cv2.imwrite(os.path.join(spath1, filename), crop)
                cv2.imwrite(os.path.join(spath2, filename), orcrop)
                cv2.imwrite(os.path.join(spath3, filename), skin)
                # cv2.imshow('', skin)
                # cv2.waitKey()

if __name__ == '__main__':
    path = r'D:\Grip_Posture\Raw_Data'
    spath1 = r'D:\Grip_Posture\Train_New\gesture1'
    spath2 = r'D:\Grip_Posture\Train_New\gesture2'
    spath3 = r'D:\Grip_Posture\Train_New\gesture3'
    run(path, spath1, spath2, spath3)

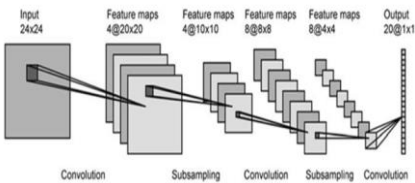
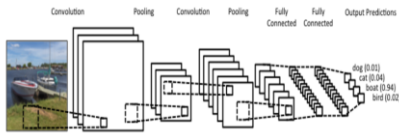
run() if not os.path.exists(path)
```

The IDE shows the script being executed, with the output 'Process finished with exit code 0' displayed in the Run console. The status bar at the bottom indicates the IDE is ready to update.

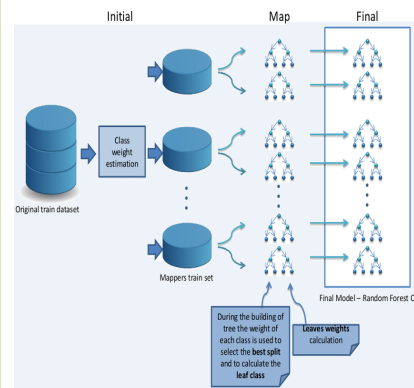
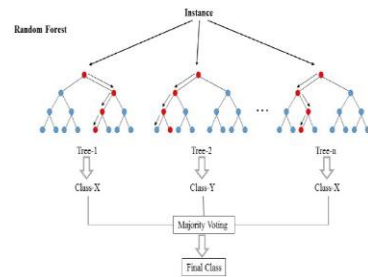
Classification

- Image classification에 (1) CNN, (2) Random forest의 2가지 기반 모델을 선정

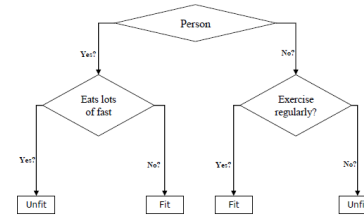
CNN (Convolutional neural network)



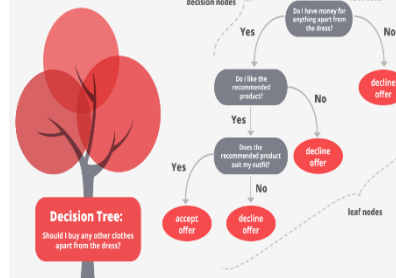
Random forest



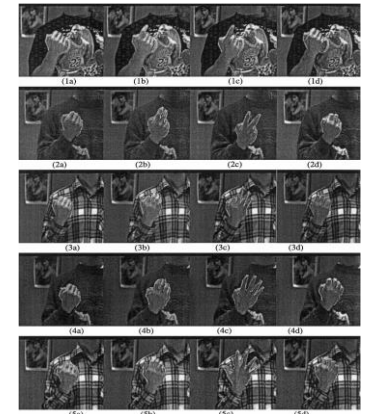
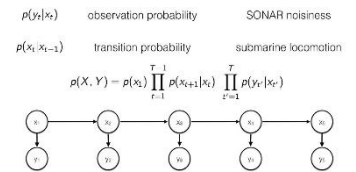
Decision tree



DECISION TREE FOR RETAILING



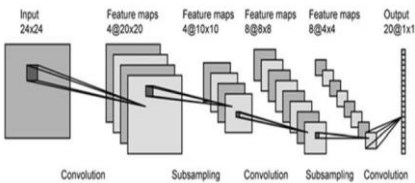
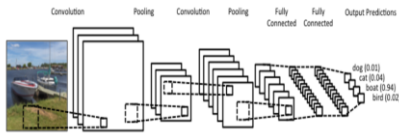
HMM (Hidden Markov Model)



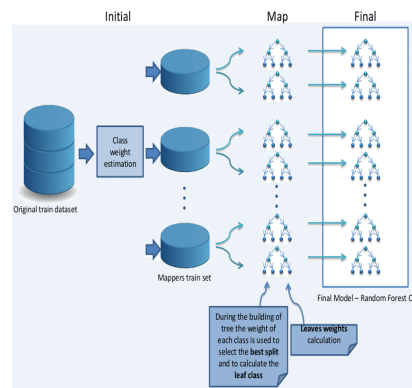
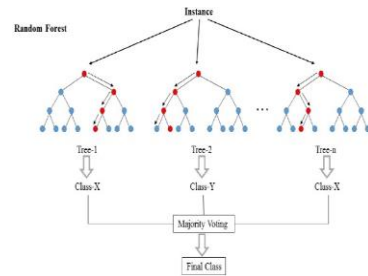
Classification

- Image classification에 (1) CNN, (2) Random forest의 2가지 기반 모델을 선정

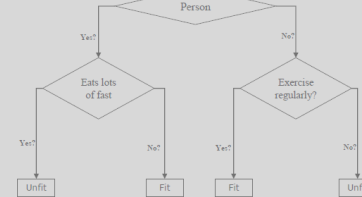
CNN (Convolutional neural network)



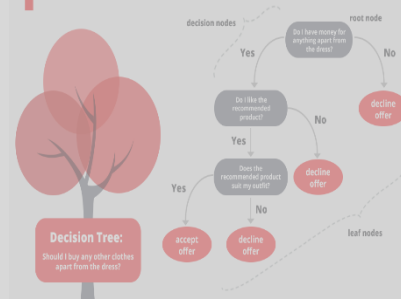
Random forest



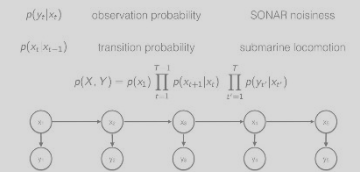
Decision tree



DECISION TREE FOR RETAILING

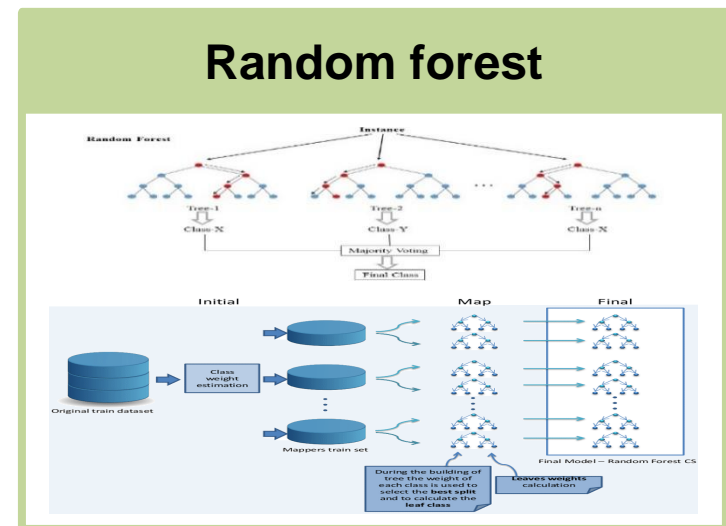
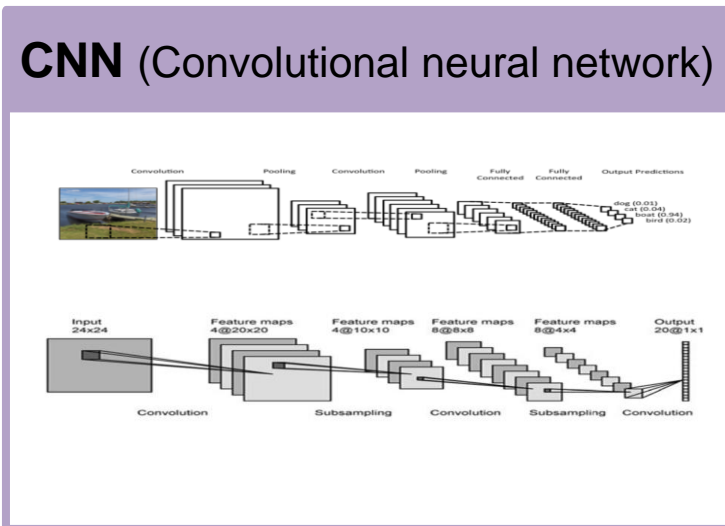


HMM (Hidden Markov Model)



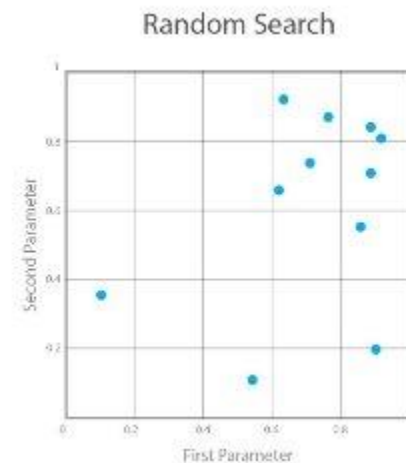
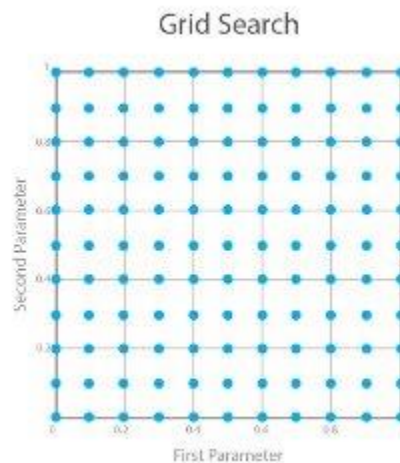
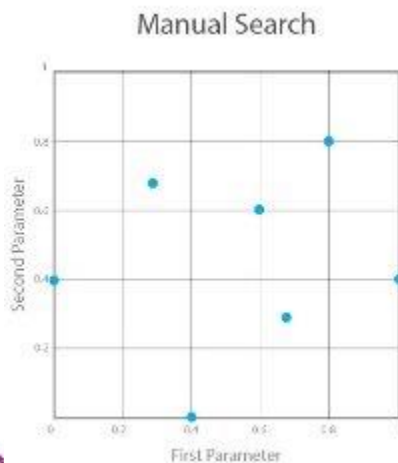
Classification

- ❑ Convolutional Neural Network (CNN)은 이미지 데이터 학습을 위한 효과적 알고리즘
 - **Layer 수 증가에 따라 정확도가 상승함**
 - 이미지 데이터 분류에 가장 일반적으로 사용되는 알고리즘
- ❑ Random forest (RF)
 - **Classification에 우수한 성능을 보여 일반적으로 사용되는 알고리즘**



Hyperparameter Tuning 알고리즘: Grid Search

- RF 활용 시 알고리즘의 정확도에 영향을 주는 매개 변수 조정
(depth of each trees, # of trees)
- Hyperparameter tuning에는 대표적으로 3가지 방법이 주로 사용됨
 - Manual search: 매개 변수를 수동/반복적으로 입력하여 찾음
 - Grid search: 매개 변수 값을 지속적으로 변화시켜 찾음
 - Random search: 매개 변수를 무작위로 검색하여 찾는 방법



Results: Random Forest 기반 분류 모델

□ 각 parameter별 값 범위

- max depth: 5 ~ 11
- Minimum samples of leaf: 1, 2, 4
- Minimum samples of splits: 1, 2, 4
- the number of estimator: 15 ~ 149

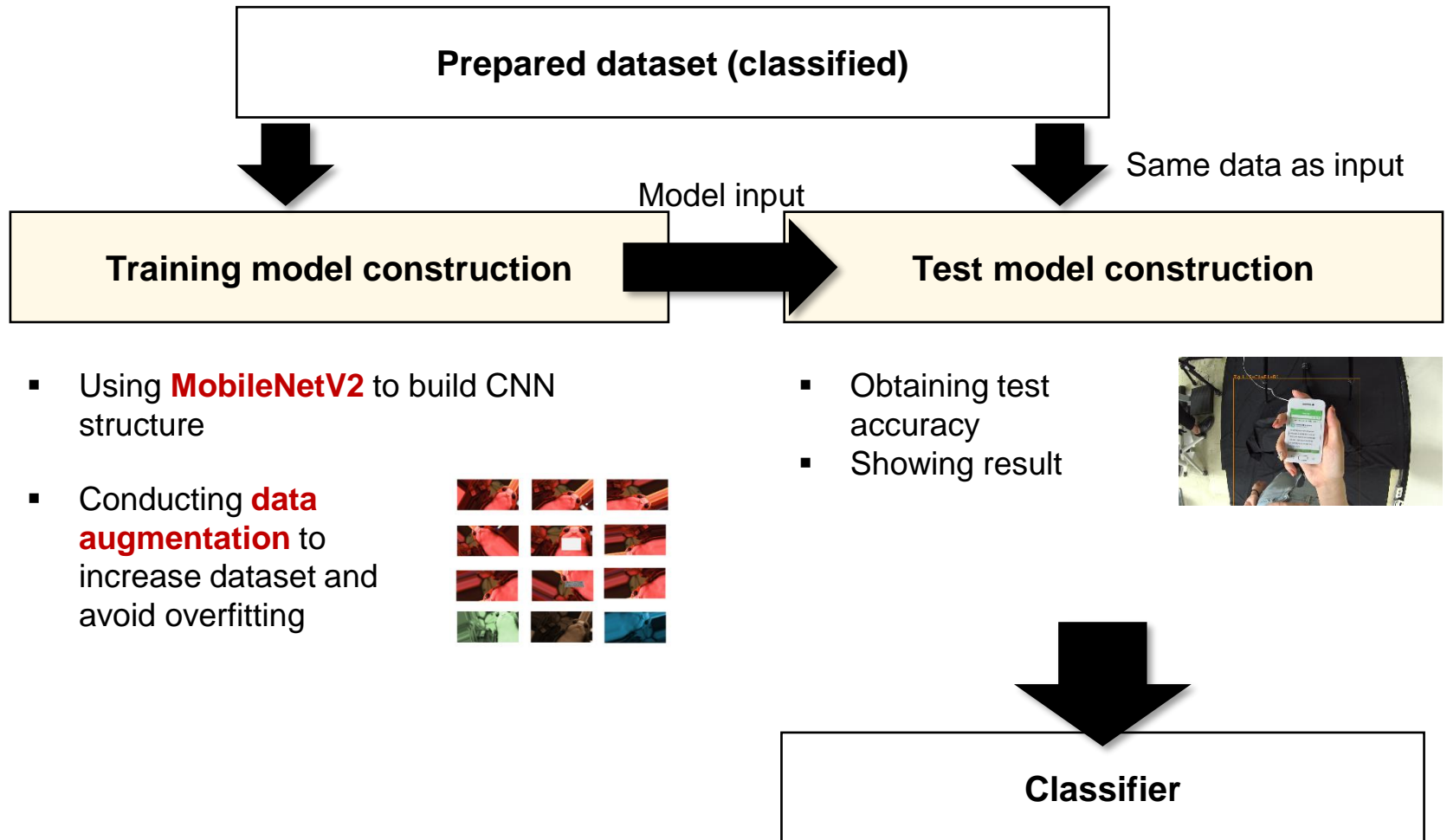
□ Best hyperparameters: max depth = 10, minimum samples of leaf = 2, minimum samples of splits = 2, the number of estimator = 135

→ test set **score: 0.857**

Example of hyperparameter tuning

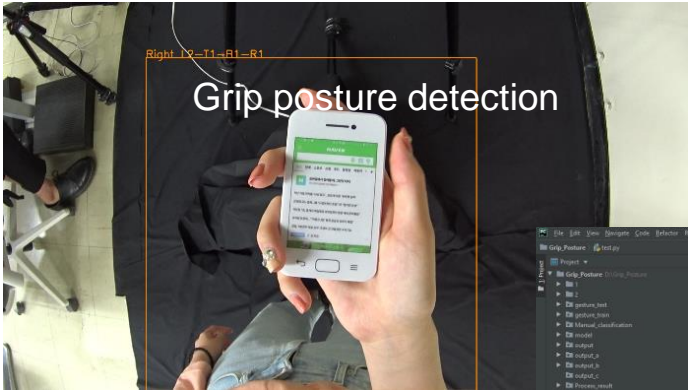
```
max_depth=5, min_samples_leaf=1, min_samples_split=2, n_estimators=15, oob_score=False, score=0.628,  
max_depth=5, min_samples_leaf=1, min_samples_split=2, n_estimators=17, oob_score=False, score=0.630,
```

CNN 기반 분류 모델: Overview



Results: CNN 기반 분류 모델

□ 개발된 classification 모델을 통해 파지 자세 이미지 구분



Classified folder

```
def show_result(img):
    predict_coord = net.predict(img)
    cv2.rectangle(img, (int(coord[0]), int(coord[1]), int(coord[2]), int(coord[3])),
                  (int(coord[4]), int(coord[5]), int(coord[6]), int(coord[7])),
                  (255, 0, 0))
    cv2.putText(img, predict_coord[0], int(coord[1]), cv2.FONT_HERSHEY_SIMPLEX,
                1.5, (255, 0, 0))
    return img

def classifier_script(img, save_path, filename):
    predict_coord = net.predict(img)
    if not os.path.exists(save_path):
        os.makedirs(save_path)
    cv2.imwrite(os.path.join(save_path, filename), img)

if __name__ == '__main__':
    path = 'C:\ProgramData\ergo\test_data'
    path = 'C:\ProgramData\ergo\test_data'
    save_path = 'C:\ProgramData\ergo\test_data'

    if not os.path.exists(save_path):
        os.makedirs(save_path)

    with TF.Session() as sess:
        handle = RestoreClassifier()
        for filename in os.listdir(path):
            if not filename.endswith('.jpg'):
                continue
            print(filename)
            img = cv2.imread(os.path.join(path, filename))
            img = handle.show_result(img)
            cv2.imwrite(os.path.join(save_path, filename), img)
            handle.classifier_script(img, save_path, filename)

# _name_ = '__main__' with if-Session() as sess: for filename in os.listdir(path)
```

Results: CNN 기반 분류 모델

- CNN model 기반 분류 accuracy (average): **0.944294**

	Correct 데이터 수	Input 데이터 수	
Right_L2-B1-R1-K1	186.0	202.0	0.920792
Right_L2-R1-K2	156.0	175.0	0.891429
Right_L2-T1-B1-R1	184.0	185.0	0.994595
Right_L3-B1-R1	207.0	212.0	0.976415
Right_L3-R1-K1	165.0	187.0	0.882353
Right_L3-R1-T1	226.0	229.0	0.986900
Right_L4-R1	158.0	165.0	0.957576

각 파지자세별 정확도

Discussion

- 본 연구는 머신러닝 기법을 사용하여 **파지 자세를 탐지/분류**하였으며 사용된 **두가지 분류 모델(CNN, RF)의 성능을 비교/평가** 하였음
- 본 연구에서 개발된 모델의 정확도는 (1) **CNN: 94.4%**, (2) **RF: 85.7%**로 파지 자세를 파악하는데 효과적인 것으로 나타남



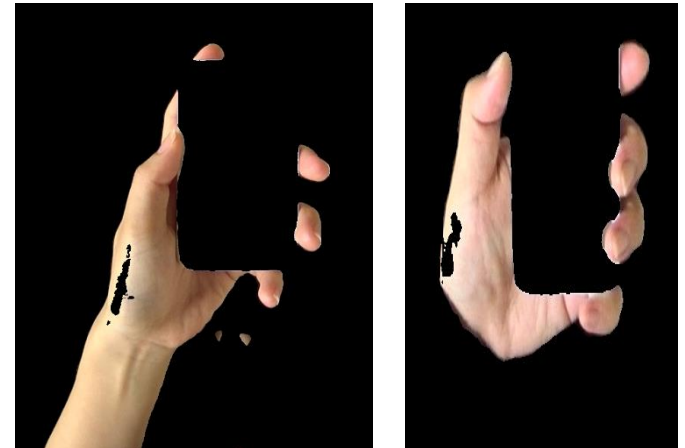
Limitation

- 데이터 전처리 시, 정확도를 높일 수 있는 방법 탐색 필요
 - 실험 참여자의 **복장이 피부 색과 비슷한 경우**, 전처리가 어려움
 - 전처리 시, 데이터의 **크기와 픽셀을 일괄적으로 통일**할 필요가 있음

Color 기반 이미지 분류 모델의 문제점 예시



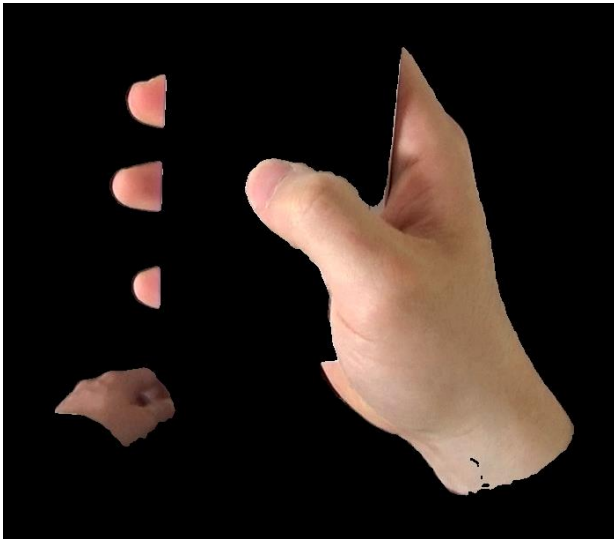
이미지의 크기가 다른 예시(동일 참여자)



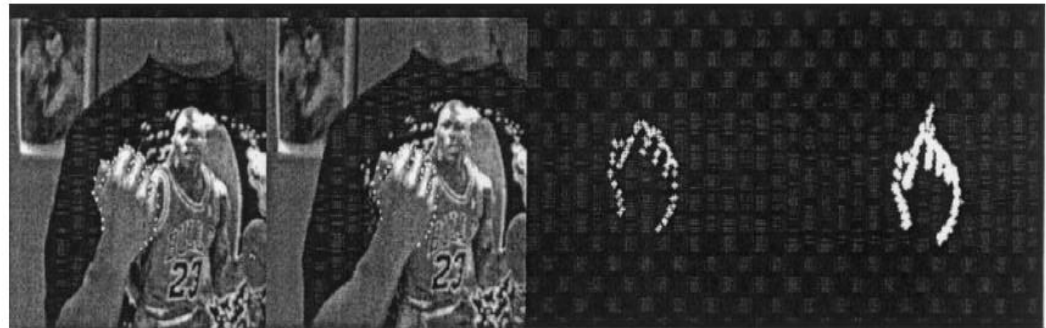
Limitation

- 본 연구에서는 사용자의 physical user interface(PUI) 사용 상황의 파지 자세만 구분하였음 ⇒ **Graphical user interface(GUI) 사용 상황 고려**
- 파지 자세 구분에 2가지 모델만 사용하여 구분함
⇒ **다양한 구분 모델 활용**

Image data when use GUI task



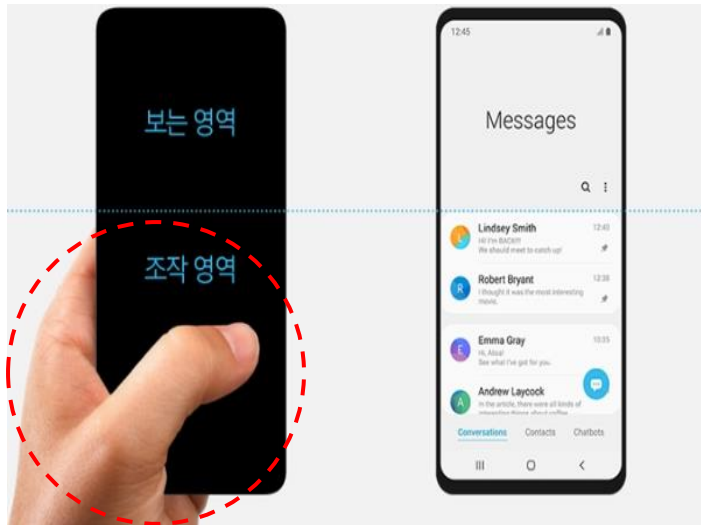
Feature extract using HMM AI technique



Future Study

- ❑ **PUI/GUI 모두 구분**할 수 있는 체계와 모델 개발
- ❑ **다른 소형가전 기기**에서도 활용될 수 있는 분류 체계와 모델 개발

GUI 사용 시의 파지 자세



다양한 소형가전 파지 예시





본 연구는 산업통상자원부의 "미래첨단 사용자편의서비스 기반조성사업"(R0004840, 2020)의 지원을 받아 수행된 연구결과입니다.